METAHEURISTIC TECHNIQUES FOR SOLVING OPTIMIZATION PROBLEMS IN GRAPH THEORY

A Synopsis

Submitted in the partial fulfilment of the requirements for the

degree of

Doctor of Philosophy

in

Mathematics

Submitted by

Yogita Singh Kardam

Dr. Kamal Srivastava	Prof. Ravindra Kumar	Prof. Sahab Dass
Supervisor	Head	Dean

Department of Mathematics
Faculty of Science
Dayalbagh Educational Institute, Dayalbagh
(Deemed University)
September, 2014

1 Introduction

Graph theory is a branch of Mathematics concerned with the analysis of structures consisting of points joined by lines. Graph theoretic models have extensive applications in many fields like computer science, operation research, business, microbiology, chemistry, medicine, sociology etc. Many optimization problems of practical interest can be represented by graphs. There exist several complicated optimization problems which can be solved using graph theoretical concepts because of its easiness to represent the real world problems. Some of these problems are travelling salesman problem, GSM (Groups Special Mobile) mobile phone networks, time table scheduling problem, graph coloring, resource allocation etc. This has generated interest in developing algorithms for graph handling. Although many of the optimization problems in graph theory are polynomial time solvable but an equally large number of them are NP-hard, i.e. no polynomial time algorithm exists which can find an optimal solution for all problem instances. Therefore, exact methods may be impractical for large scale problem instances. For most of their applications, feasible solutions with an almost optimal cost are sufficient. Consequently, there is a great interest in developing efficient methods that yield optimal or near optimal solutions in acceptable computational time. A class of such methods is known as Metaheuristics which are described in the next section.

2 Metaheuristics

In recent years, metaheuristics are emerging as successful alternatives to classical approaches for solving optimization problems. These are powerful search methods which can be efficiently used in providing satisfactory solution to large and complex problems. Metaheuristic techniques [36] are high-level frameworks which utilize heuristics in order to find solutions to combinatorial optimization problems at a reasonable computational cost. Such a problem may have been classified as NP-complete or NP-hard, or be a problem for which a polynomial time algorithm is known to exist but is not practical. They are alternate methods to solve optimization problems though they do not provide a theoretical guarantee on results [9]. However they do provide near-

optimal solutions for many problem instances that arise in practice. Besides, they have a wide range of applicability which is the most appealing aspect of metaheuristics.

Metaheuristics are the general search procedure whose principles allow them to escape the trap of local optimality using heuristic designs. These are strategies that guide the search process. The goal of these is to efficiently explore the search space in order to find optimal or near optimal solutions. The techniques which constitute metaheuristic algorithms range from simple local search procedures to complex learning processes.

A definition of metaheuristic based on intensification (exploitation) and diversification (exploration) as well as on the iterative nature of the search space can be given as [29]:

"A metaheuristic is an iterative generation process which guides a subordinate heuristic by combining intelligently different concepts for exploring and exploiting the search space, learning strategies are used to structure information in order to find efficiently near-optimal solutions."

Block diagram of a metaheuristic [36] is shown below:

Algorithm: Abstract algorithmic framework for metaheuristics

```
Create one or several start solutions e.g. by problem specific heuristic(s)
while termination criterion not satisfied do
    if intensify then
        Create new solution by intensification step;
    else
        Create new solution by diversification step;
    end
    Update best found solution (if necessary);
end
return Best found solution;
```

There are several metaheuristics already known in the literature. Metaheuristics can be categorized as population based (e.g., genetic algorithm, particle swarm optimization etc.) and

trajectory based (e.g. tabu search). A population based metaheuristic (P-metaheuristic) involves a population of solutions and a trajectory based metaheuristic (T-metaheuristic) manipulates and transforms a single solution during the search. These two are complementary in the sense that a P-metaheuristic *explores* solutions in the whole search space and a T-metaheuristics *exploits* the solutions, leading to intensification of search in local regions. T-metaheuristic is essentially a local search method wherein an intensive search of the solution space is performed by moving at each step from the current solution to another promising solution in its neighborhood. Some popular local search methods are Simulated Annealing (SA), Tabu Search (TS) and Variable Neighborhood Search (VNS). Population search consists of maintaining a pool of good solutions and applying some intelligent scheme on them in order to produce hopefully better solutions. Examples of these are Genetic Algorithms (GA), Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), Artificial Bee Colony (ABC) and adaptive memory procedures [21].

Some of the metaheuristics mentioned above are briefly described in the following paragraphs:

Genetic Algorithms (GAs) were invented by Holland [18] in the early 1970's. GA is a search heuristic that mimics the process of natural evolution. This heuristic is routinely used to generate useful solutions to optimization and search problems. GA belongs to the larger class of Evolutionary Algorithms (EA), which generate solution to optimization problems using techniques inspired by natural evolution such as inheritance, selection, crossover, and mutation.

Ant Colony Optimization (ACO) is inspired by the foraging behavior of real ants. Due to their foraging behavior, ants find the shortest paths between their nest and food source. While walking, ants lay down on the ground a chemical substance called pheromone, forming in this way, a pheromone trail which is used for indirect communication among ants [34]. ACO is a metaheuristic in which a colony of artificial ants cooperates in finding good solutions to difficult discrete optimization problems. An artificial ant is a stochastic constructive procedure that incrementally builds a solution by adding opportunely defined solution components to a partial solution under construction. Therefore, ACO can be applied to any combinatorial optimization problem for which a construction heuristic can be defined.

Particle swarm optimization (PSO) [36] is another population based metaheuristic that optimizes a problem by maintaining a population of candidate solutions, called *particles*. The population is also sometimes called a *swarm* and its behavior is similar to that of birds and fishes in search of food. The particle has a *position* and *velocity* and it adjusts its position according to its current velocity, the best position it has achieved so far (*pbest*) and the best position of the swarm (*gbest*). Thus, the movements of particles are guided by the best found positions in the search-space, which are continually updated as better positions are found by the particles.

The Artificial Bee Colony (ABC) algorithm was introduced by Karaboga [22] for numerical optimization problems. It is a recent swarm based metaheuristic technique that is used to solve hard optimization problems. It is based on the intelligent foraging behavior of honey bee swarm [34]. ABC as an optimization tool provides a population-based search procedure in which individuals called food positions are modified by the artificial bees with time and the bee's aim is to discover the places of food sources with high nectar amount and finally the one with the highest nectar. In ABC system, artificial bees fly around in a multidimensional search space and some (employed and onlooker bees) choose food sources depending on the experience of themselves and their nest mates, and adjust their positions. Some (scouts) fly and choose the food sources randomly without using experience. If the nectar amount of a new source is higher than that of the previous one in their memory, they memorize the new position and forget the previous one. Thus, ABC system combines local search methods, carried out by employed and onlooker bees, with global search methods, managed by onlookers and scouts, attempting to balance exploration and exploitation process.

One of the most recent optimization metaheuristics - Migrating Birds Optimization (MBO) algorithm, is developed by Duman et al. [11]. It is a neighborhood search technique inspired from the V formation flight of the migrating birds which is proven to be an effective formation in energy minimization. It starts with a number of initial solutions. Starting with the first solution (corresponding to the leader bird), and progressing on the lines towards the tails, each solution is tried to be improved by its neighbor solutions. If the best neighbor solution brings an improvement, the current solution is replaced by that one. There is also a benefit mechanism for the solutions (birds) from the solutions in front of them. A solution evaluates the number of its own neighbors and number of best neighbors of the previous solution and is considered to be

replaced by the best of them. Once all solutions are improved (or tried to be improved) by neighbor solutions, this procedure is repeated a number of times (tours) after which the first solution becomes the last, and one of the second solutions becomes first and another loop starts. MBO algorithm was tested for the quadratic assignment problem instances arising from a real life problem and has been shown to give better results when compared with the other techniques like GA, SA, PSO etc.

Simulated Annealing (SA) is a local optimization method introduced by Kirkpatrick et al. [23] and Cerny [8]. It is based on a model developed by Metropolis et al. [28] for simulating the physical annealing process, where particles of a solid arrange themselves into a thermal equilibrium. The fundamental idea of SA is to allow moves resulting in solutions of worse quality than the current solution (‘‘*uphill moves*’’) in order to escape from local optima. The probability of doing such a move is decreased during the search process. The algorithm uses a pre-defined neighborhood structure on the search space. A control parameter which is called ‘‘*temperature*’’ in analogy to the physical annealing process governs the search behavior. In each iteration, a neighbor solution to the current solution is computed. If this neighbor solution has a better objective function value than the current solution, the neighbor solution is accepted, i.e. the current solution is replaced by this neighbor solution. If, on the other hand, neighbor solution has a worse objective function value than the current one, then neighbor solution is only accepted with a certain probability depending on (i) the difference of the objective function values in current and neighbor solutions, and (ii) the temperature parameter. In SA, intensification is provided by the local search nature of the algorithm, while diversification is produced by the presence of ‘‘*uphill moves*’’ that are possible for non-zero temperature [2].

Variable Neighborhood Search (VNS) is a metaheuristic methodology for solving optimization problems based on a systematic change of neighborhood structures, without guaranteeing the solution's optimality. It was developed by Hansen and Mladenovic [20]. VNS is a local search procedure, but contrary to SA, it does not use a single neighborhood structure, but instead a hierarchy of more narrow as well as of broader neighborhood definitions is used. From an ‘‘incumbent’’ solution x, the procedure jumps to a random neighbor solution x' contained in one of these neighborhoods (a step called *shaking*), improves the solution x' by local search according to the smallest pre-defined neighborhood, which yields a solution x", and accepts or

rejects x" depending on whether it is better than x or not. In the case of rejection, the current neighborhood size for the shaking step is increased by one, in the case of acceptance, it is re-set to the smallest value [2]. There are several variations of the VNS which includes Variable Neighborhood Descent (VND), Reduced VNS (RVNS), Skewed VNS (SVNS) and Variable Neighborhood Decomposition Search (VNDS) etc.

Tabu Search (TS) was proposed by Fred Glover [17] to allow hill climbing to overcome local optima. The basic principle of TS is to pursue the search whenever a local optimum is encountered by allowing non-improving moves. Cycling back to previously visited solutions is prevented by the use of memories, called tabu lists, that record the recent history of the search. In this metaheuristic, intensification is provided, like in SA, by the local search mechanism, while diversification is given by the use of tabu lists.

3 Some Graph Theoretic Optimization Problems

Many real-world situations can conveniently be described by means of a diagram consisting of a set of points together with lines joining certain pairs of these points. A mathematical abstraction of these situations gives rise to the concept of graph. Thus, graphs are mathematical structures used to model pairwise relations between objects from a certain collection i.e. a graph is a set of objects called vertices, points or nodes connected by links called lines or edges. The origin of graph theory started with the problem of Konigsberg bridge, in 1735. Now it is rapidly moving into the mainstream of Mathematics mainly because of its applications in diverse fields.

As mentioned earlier, many phenomena from various areas can be modeled on graphs and there exist a large class of problems where certain objective function has to be optimized. We present here some graph optimization problems which we intend to take up in this work as they are important from applications point of view but are NP hard for general graphs. Optimal solutions for these problems do exist in literature but only for some particular classes of graphs. Moreover, in the literature reviewed by us, only a few metaheuristics are available for these problems.

We first give some definitions and relevant preliminaries.

A *graph* $G(V, E)$ is a set of vertices $V = \{v_1, v_2,\ldots,v_n\}$ together with a set of edges $E = \{e_1, e_2,\ldots,e_m\}$, such that each edge e_k is a pair of vertices (v_i, v_j) and is denoted by v_iv_j. The *neighborhood* of a vertex v is defined as $N(v) = \{ u \in V / uv \in E \}$. A graph G is *directed* or is a *digraph* if there is a function f which assigns an ordered pair of vertices to each edge e_k i.e. for each $e_k \in E$, $\exists\, u, v \in V$ such that $f(e_k) = (u,v) \in V \times V$. A digraph is thus denoted as $DG(V, E, f)$. If a graph is not directed, then it is said to be undirected. A *uv-path* in a graph G is an alternating sequence $S = \{ u = v_0, e_1, v_1, e_2,\ldots,e_k, v_k = v\}$ such that $e_i = (v_{i-1}, v_i)$ $\forall i = 1, \ldots, n$ and each vertex in the sequence is distinct. Sequence S is a *cycle* in G if all the vertices in it are distinct except that $v_0 = v_k$. A graph without any cycle is called *acyclic*. A graph G is *connected* if it has a uv-path whenever $u, v \in V$, otherwise G is *disconnected*. A *tree* is a connected acyclic graph. A graph $G(V, E)$ together with a map $\varphi_E : E \rightarrow \mathcal{R}^+$ is an *edge weighted* graph denoted by $G(V, E, \varphi_E)$ and if a map $\varphi_V : V \rightarrow \mathcal{R}^+$ is associated with G then $G(V, E, \varphi_V)$ is said to be a *vertex weighted* graph. Throughout, the graphs are assumed to be undirected unless stated otherwise.

3.1 Dominating Tree Problem (*DTP*)

Given a connected and edge-weighted graph, the dominating tree problem (*DTP*) seeks on this graph a tree with minimum total edge weight such that each vertex of the graph is either in this tree or adjacent to a vertex in this tree [34]. This tree is termed as a dominating tree (DT) and the vertices of DT are called dominating vertices. Remaining vertices of V are the non-dominating vertices. It is widely used in the area of wireless sensor networks. A solution to this provides a virtual backbone for routing.

3.2 Minimum Weighted Edge Dominating Set Problem *(MWEDSP)*

For a graph $G(V, E)$ an Edge Dominating Set (EDS) is a subset of edges where each edge of E is either in EDS or adjacent to at least one edge of EDS. The minimum edge dominating set problem (MEDSP) is to find an EDS of minimum cardinality. For an edge weighted graph, the weighted version of MEDS seeks to find an EDS with minimum total weight [1]. Mathematically, given an edge weighted graph $G(V, E, \varphi_E)$, *MWEDSP* is to

find $\min_{\forall\, EDS} \sum_{e_i \in EDS} \varphi(e_i)$. This problem has applications in operation research, resource allocation, network routing and coding theory.

3.3 Minimum Edge Total Dominating Set Problem *(METDSP)*

This problem is a variant of MEDSP. Let $G(V, E)$ be a graph without any isolated edge. A subset F of E is an Edge Total Dominating Set (ETDS) of G if each edge in E is adjacent to at least one edge in F. The minimum edge total dominating set problem (*METDSP*) seeks to find ETDS of minimum cardinality. The edge total domination number of G is the minimum size among all ETDSs of G [33].

3.4 Vertex Separation Problem *(VSP)*

For a graph $G(V, E)$, a linear layout φ of G is a bijection $\varphi: V \rightarrow \{1,2,\ldots,n\}$. Hence $\varphi(u)$ denotes the position or label of vertex u in layout φ. Let Ω be the set of all layouts φ. Define, $L(p, \varphi, G) = \{u \in V : \varphi(u) \leq p\}$ and $R(p, \varphi, G) = \{u \in V : \varphi(u) > p\}$. The *Cut*-value at position $p \in \{1,2,...,|V|-1\}$ for layout φ is $Cut(p, \varphi, G) = |\{u \in L(p, \varphi, G) / R(p, \varphi, G) \cap N(u) \neq \emptyset\}|$. The Vertex Separation (*VS*) of G for a layout φ is the maximum of *Cut*-value among all positions in layout φ *i.e.* $VS(\varphi, G) = \max_{\forall p} Cut(p, \varphi, G)$. The Vertex Separation Problem (*VSP*) consists of finding a layout $\varphi \in \Omega$ which minimizes *VS* in graph G i.e. $\min_{\forall\, \varphi \in \Omega} VS(\varphi, G)$. VSP has applications in VLSI, computer language compiler design and graph drawing.

3.5 Feedback Vertex Set Problem (*FVSP*)

In a directed graph $DG(V, E, f)$, a feedback set is a subset of V which when removed from G makes it acyclic. The Feedback Vertex Set Problem consists in making a given directed graph acyclic by removing as few vertices as possible [15]. In other words, given a directed graph, the goal of *FVSP* is to find a feedback set with minimum cardinality. For a given undirected, vertex weighted graph $G(V, E, \varphi_V)$, weighted feedback vertex set problem (WFVSP) is to find a feedback vertex set with minimum total weight. *FVSP* has applications in many areas such as VLSI design, program verification, deadlock resolution, and Bayesian inference.

4 Literature Review

Exact results for some classes of graphs are available in literature for the problems mentioned above. Some approximation algorithms have been developed and researchers have also focused on designing metaheuristics for these problems. Since the focus of this work is on designing and implementing metaheuristics for solving these problems, so we have surveyed mainly the heuristic algorithms which have been developed for these problems. Besides, the exact results available for some classes will be used to test the performance of our algorithms. Therefore, in the following sections we give an outline of the relevant work done on these problems so far.

4.1 Dominating Tree Problem

DTP has important applications and is widely used in the area of wireless sensor networks even though it has been shown to be NP-hard problem [34]. It was introduced by Zhang et al. [37] and Shin et al. [31] with the aim of minimizing energy consumption of a routing scheme, considering energy consumption by each edge rather than by nodes. Guha et al. [19] have presented two polynomial time approximation algorithms for connected dominating sets. Zhang et al. [37] proposed a heuristic to approximate the solution with low time complexity and provided a distributed algorithm for practical implementation. Recently, Sunder et al. [34] proposed a new heuristic approach and two swarm intelligence techniques, namely an Artificial Bee Colony (ABC) algorithm and an Ant Colony Optimization (ACO) algorithm for *DTP*. There is still a wide scope for other metaheuristics namely- memetic algorithm, Variable neighborhood search etc, which may require designing of problem dependent neighborhood structures and construction procedures in order to generate good initial solutions.

4.2 Minimum Weighted Edge Dominating Set Problem

The Minimum Edge Dominating Set (MEDS) problem is classified as NP-hard combinatorial problem [1], and thus cannot be solved exactly in polynomial time. Edge Dominating Set (EDS) problem for the unweighted graphs was introduced by Garey and Johnson [16]. Yannakakis and Gavril [35] proved that it is NP-hard even for planar or bipartite graphs of maximum degree 3. Many approximation and exact algorithms have been developed for this problem. There are many algorithms which have been proposed for solving Minimum Weighted Edge Dominating

Set (*MWEDS*). Although these algorithms guarantee for the optimality of solutions they find, they may fail to give a solution within reasonable time for large instances. There is very little work available in literature which uses meta-heuristic techniques for solving *MWEDS* problem. Recently, Aziz et al. [1] proposed a metaheuristic approach based on genetic algorithm and local search to solve *MWEDS*.

4.3 Minimum Edge Total Dominating Set Problem

Kulli and Patwari [24] introduced the concept of edge total dominating set (*ETDS*) in 1999 but this problem remains almost unstudied so far. Recently, Song et al. [33] have obtained exact value of edge total domination number for multipartite graphs using the concept of maximal matching. No heuristic / algorithm for the problem exists in literature surveyed by us so far.

4.4 Vertex Separation Problem

VSP was originally motivated by the general problem of finding good separators for graphs [27], and has applications in algorithms for VLSI design [25]. *VSP* is an NP-hard optimization problem [9]. This problem has been generally tackled with exact approaches, presenting polynomial time algorithms to obtain the optimal solution for specific types of graphs. A linear time algorithm to find the optimal layout of a tree is proposed in [32]. Polynomial time algorithms to compute the vertex separation of permutation graphs [4] and cographs [3] have also been developed. *VSP* can be solved in polynomial time for unicyclic graphs [12], permutation graphs, starlike graphs, and partial k-trees for a fixed k [9]. Recently, a pure 0-1 optimization model and a metaheuristic algorithm based on variable neighborhood search methodology for *VSP* has been proposed by Duarte et al. [10]. However, no other heuristic or metaheuristic algorithm has been developed for *VSP* to find good solutions in short computing times.

4.5 Feedback Vertex Set Problem

The Feedback Vertex Set Problem is NP-complete [6]. This problem has been extensively studied, especially from the standpoint of approximation algorithms. In contrast to the numerous approximation schemes, relatively only a few exact algorithms have been proposed [14]. Lin and

Jou [26] have developed branch and bound algorithm for this problem. This problem is polynomial time solvable on particular class of graphs like: k-diamond graphs, permutation graphs, interval graphs, convex bipartite graphs, reducible flow graphs and co-comparability graphs [6]. On the other hand, very few heuristics have been proposed for *FVSP*. Only heuristics available in the literature are greedy heuristics and the Greedy Randomized Adaptive Search procedure (GRASP), proposed by Pardalos et al. [30] and Festa et al. [15] respectively. Recently, Galinier et al. [15] suggested a new representation of the solution based on a property of acyclic graphs and designed a local transformation (neighborhood) that changes a feedback set into a new one and developed a Simulating Annealing (SA) algorithm based on this neighborhood. No population based metaheuristics have been designed for the problem which may also require designing of new construction heuristics and local search strategies for the improvement of solutions. A weighted version of this problem also exists in literature. The two metaheuristics eXploring Tabu Search (*XTS*) [5] and Iterated Tabu Search (*ITS*) [7] have been proposed for this problem. Very recently, Carrabs et al. [6] proposed a memetic algorithm, outperforming the effectiveness of the two metaheuristics.

5 Objectives

In this work, we will explore and identify applications of metaheuristics to the optimization problems among the Dominating Tree Problem (*DTP*), Minimum Weighted Edge Dominating Set Problem (*MWEDSP*), Minimum Edge Total Dominating Set Problem (*METDSP*), Vertex Separation Problem (*VSP*) and Feedback Vertex Set Problem (*FVSP*) for general graphs. This may involve developing problem specific construction heuristics in order to generate good initial solutions specially needed for the population based metaheuristics. Problem dependent genetic operators such as crossover, mutation etc. may also be designed as existing operators usually are not efficient enough to solve specific problems. Local improvement strategies and neighborhood structures will be another focus area as they play a key role in the performance of a metaheuristic. The developed construction heuristics / operators / techniques will be statistically evaluated on benchmark problems / random graphs and will also be tested on standard graphs with known results. Some other optimization problems concerned with graphs having real life applications may also be explored.

References

[1] Aziz S. N. A., Hedar A. R. and Sewisy A. A., Memetic Algorithm with Filtering Scheme for the Minimum Weighted Edge Dominating Set Problem, International Journal of Advanced Research in Artificial Intelligence, vol-2, no.8, pp. 44-49, 2013.

[2] Bianchi L., Dorigo M., Gambardella L. M. and Gutjahr W. J., A survey on metaheuristics for stochastic combinatorial optimization, Natural Computing, vol-8, pp. 239-287, 2009.

[3] Bodlaender H. L. and Mohring R. H., The pathwidth and treewidth of cographs, SIAM Journal on Discrete Mathematics, vol-6, no.2, pp. 181–188, 1993.

[4] Bodlaender H. L., Kloks T. and Kratsch D., Treewidth and pathwidth of permutation graphs, SIAM Journal on Discrete Mathematics, vol-4, no.4, pp. 606-616, 1995(b).

[5] Brunetta L., Maffioli F. and Trubian M., Solving the feedback vertex set problem on undirected graphs, Discrete Applied Mathematics, vol-101, pp. 37-51, 1999.

[6] Carrabs F., Cerrone C. and Cerulli R., A Memetic Algorithm for the Weighted Feedback Vertex Set Problem, To appear on Networks, 2014.

[7] Carrabs F., Cerulli R., Gentili M. and Parlato G., A tabu search heuristic based on k-diamonds for the weighted feedback vertex set problem, Lecture Notes in Computer Science, vol-6701, pp. 589-602, 2011.

[8] Cerny V., A Thermodynamical Approach to the Traveling Salesman Problem: An Efficient Simulated Algorithm, Journal of Optimization Theory and Applications, vol-45, pp. 41-55, 1985.

[9] Diaz J., Petit J. and Serna M., A survey of graph layout problems, ACM Computing Surveys, vol-34, pp. 313-356, 2002.

[10] Duarte A., Escudero L. F., Marti R., Mladenovic N., Pantrigo J. J. and Oro J. S., Variable Neighborhood Search for the Vertex Separation Problem, Computer and Operation Research, vol-39, no. 12, pp. 3247-3255, 2012.

[11] Duman E., Uysal M. and Alkaya A. F., Migrating Birds Optimization: A new metaheuristic approach and its performance on quadratic assignment problem, Information Sciences, vol-217, pp. 65-77, 2012.

[12] Ellis and Markov, Computing the vertex separation of unicyclic graphs, INFCTRL: Information and Computation (formerly Information and Control), vol-192, 2004.

[13] Festa P., Pardalos P. M. and Resende M. G. C., Algorithm 815: FORTRAN Subroutines for Computing Approximate Solutions of Feedback Set Problems using GRASP, ACM Transactions on Mathematical Software, vol-27, no. 4, pp. 456-464, 2001.

[14] Festa P., Pardalos P. M. and Resende M. G. C., Feedback set problems. In: Floudas, C. A., Pardalos, P. M. (eds.), Encyclopedia of optimization, pp. 1005–1016. Springer, Heidelberg (2009).

[15] Galinier P., Lemamou E. and Bouzidi M. W., Applying local search to the feedback vertex set problem, Journal of Heuristics, vol-19, pp. 797-818, 2013.

[16] Garey M. R. and Johnson D. S., Computers and intractability, vol-174, freeman New York, 1979.

[17] Glover F., Future paths for integer programming and links to artificial intelligence, Computers and Operations Research, vol-13, pp. 533-549, 1986.

[18] Goldberg D. E., Genetic Algorithms, Pearson Education, New Delhi, 1989.

[19] Guha S. and Khuller S., Approximation Algorithms for Connected Dominating Sets, Algorithmica, vol-20, pp. 374-387, 1998.

[20] Hansen P. and Mladenovic N., Variable neighborhood search: Principles and applications, European Journal of Operation Research, vol-130, pp. 449–467, 2001.

[21] Hertz A. and Widmer M., Guidelines for the use of meta-heuristics in combinatorial optimization, European Journal of Operational Research, vol-151, pp. 247-252, 2003.

[22] Karaboga D., An idea based on honey bee swarm for numerical optimization, Technical report tr06, Computer Engineering Department, Erciyes University, Turkey, 2005.

[23] Kirkpatrick S., Gellatt Jr. C. D. and Vecchi M. P., Optimization by simulated annealing, Science, vol-220, pp. 671–680, 1983.

[24] Kulli V. R. and Patwari D. K., On the edge total domination number of a graph, Proceedings of the Symposium on Graph Theory and Combinatorics (Cochin 1991), Publication, 21, Centre for Mathematical Sciences, Trivandrum, pp. 75-81, 1991.

[25] Lieserson C. E., Area-efficient graph layouts (for VLSI), In Proceedings of the 21st Annual Symposium on Foundations of Computer Science, pp. 270-281, 1980.

[26] Lin H. M. and Jou J. Y., Computing minimum feedback vertex sets by contraction operations and its applications on CAD, In: (ICCD ’99) International Conference on Computer Design, pp. 364–369, 1999.

[27] Lipton R. J. and Tarjan R. E., A separator theorem for planar graphs, SIAM Journal on Applied Mathematics, vol-36, pp. 177-189, 1979.

[28] Metropolis N., Rosenbluth A., Rosenbluth M., Teller A. and Teller E., Equation of state calculations by fast computing machines, The Journal of Chemical Physics, vol-21, pp. 1087–1092, 1953.

[29] Osman I. H. and Laporte G., Metaheuristics: A bibiliography, Annals of Operation Research, vol-63, pp. 513-623, 1996.

[30] Pardalos P. M., Qian T. and Resende M. G. C., A Greedy Randomized Adaptive Search Procedure for the Feedback Vertex Set Problem, Journal of Combinatorial Optimization, vol-2, pp. 399-412, 1999.

[31] Shin I., Shen Y. and Thai M. T., On approximation of dominating tree in wireless sensor networks, Optimization Letters, vol-4, pp. 393-403, 2010.

[32] Skodinis K., Construction of linear tree layouts which are optimal with respect to vertex separation in linear time, Journal of Algorithms, vol-47, no.1, pp. 40–59, 2003.

[33] Song W., Miao L., Wang H. and Zhao Y., Maximal matching and edge domination in complete multipartite graphs, International Journal of Computer Mathematics, vol-91, no.5, pp. 857-862, 2014.

[34] Sundar S. and Singh A., New heuristic approaches for the dominating tree problem, Applied Soft Computing, vol-13, pp. 4695-4703, 2013.

[35] Yannakakis M. and Gavril F., Edge dominating sets in graphs, SIAM Journal on Applied Mathematics, vol-38, no. 3, pp. 364–372, 1980.

[36] Zapfel G., Braune R. and Bogl M., Metaheuristic Search Concepts, Springer-Verlag Berlin Heidelberg, 2009.

[37] Zhang N., Shin I., Li B., Boyaci C., Tiwari R. and Thai M. T., New Approximation for Minimum-Weight Routing Backbone in Wireless Sensor Network, Lecture Notes in Computer Science, vol-5258, Springer-Verlag, Berlin, pp. 96-108, 2008.